2023 2nd International Conference on Futuristic Technologies (INCOFT)
Karnataka, India. Nov 24-26, 2023

Play Store App Rating Prediction using Machine Learning & Deep Learning Regression Techniques

Bhramara Bar Biswal
Computer Science and Engineering
Gandhi Institute of Engineering and Technology
Gunupur, India
bhramarabarbiswal@giet.edu

Satya Narayan Das
Computer Science and Engineering
Gandhi Institute of Engineering and Technology
Gunupur, India
sndas@giet.edu

Satyanarayan Sahu
Computer Science and Engineering
Gandhi Institute of Engineering and Technology
Gunupur, India
satyanarayansahu@giet.edu

Shobhan Banerjee
Department of Engineering Technology
Birla Institute of Technology and Science
Pilani, India
shobhanbanerjee3@gmail.com

2023 2nd International Conference on Futuristic Technologies (INCOFT) | 979-8-3503-0884-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/INCOFT60753.2023.10425706

***Abstract*—The rating & review of an app in the App Store or Play Store plays an essential role for end users to get information about the app based on other people's experiences with that app. The reviews might be context-based, but the ratings give a firsthand impression to the users, simply looking at scores out of five. In this paper, we have performed multivariate analysis on the Google Play Analysis Dataset to determine the critical features and prune off the redundant ones, followed by which we compared the accuracy of various regression models in predicting the ratings of various apps in the dataset.**

Keywords—Rating prediction, Regression, Multivariate Analysis, Feature Importance, Artificial Neural Network

I. Introduction

The overall performance of an app is voted on by end-users through their ratings and reviews in the store. The reviews are nothing but a justification for the rating being posted out of five by a user. Analysis of reviews is an NLP task and can't be generated in a straightforward manner. Neither it is required to generate reviews, as the first-hand impression of an app is given by its rating on a scale of zero to five. In order to predict the rating, multiple attributes and their relevance need to be analyzed based on which the rating can be predicted.

Authors in [1] proposed a model performing sentiment analysis on popular apps using a Support Vector Machine. A comparative analysis of various classical ML models with a multi-layer perceptron was performed in [2] for classifying the star rating an app could potentially acquire. In [3], the authors have performed a statistical descriptive analysis of the Google Play Store Mobile Application data using Machine Learning. A security-based ranking scheme has been proposed by the authors in [4], which has been referred to as SERS. In [5] the authors have worked on an unsupervised clustering algorithm for multidimensional data, based on a segregate and integrate method.

Authors in [6] have performed sentiment analysis on the reviews given by the end users to predict numeric ratings of apps in the Google Play Store. Authors in [7] proposed a technique to predict the success or failure of an Android app, even before it is uploaded to the store. The relationship between attributes has been studied & visualized after scraping the Google Play Store data by the authors [8]. Authors in [9] have studied the indicators that affect the reviews, ratings & the number of downloads thoroughly through an EDA on the data. The dataset used for this paper is the Google Play Store Analysis dataset that we downloaded from Kaggle [10] published by Gautham Prakash.

In this paper, we have performed a regression task, (unlike previous works where people converted the continuous-valued targets to categorical targets to perform classification) on the same dataset [10], where our target attribute is the rating based on other corresponding features for a given app. The exploratory data analysis, statistical description, correlation analysis & data preprocessing has been described in detail in [2]. After extracting the relevant features and normalizing them, we fed these values to various ML regression models, followed by which an artificial MLP was also used to feed the data and train the model to predict the ratings.

II. Data Description

The ratings range from 0 to 5 with a mean of 4.17 for a given app. The whisker plot in Figure 1 shows the data rating distribution for the whole dataset.

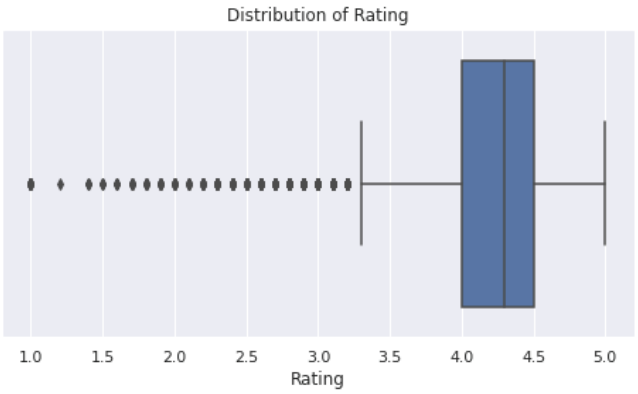

Fig. 1. Rating distribution

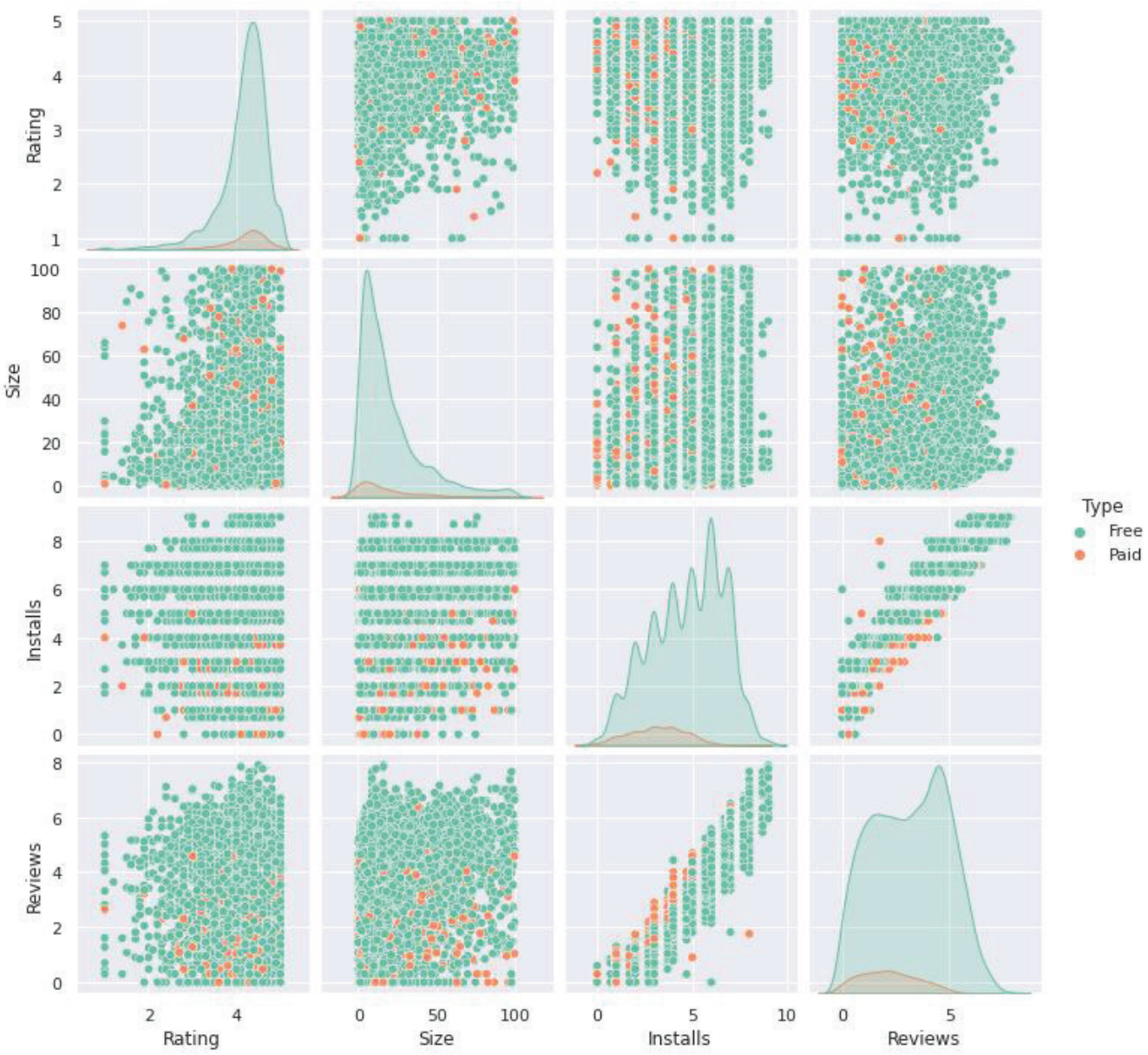

Fig. 2. Correlation Plot

The pair plot in Figure 2 shows the correlation between four attributes – Rating, Size, Installations & reviews. The orange data points represent the free apps whereas the green ones represent the paid apps. From his figure, we can infer that there is a strong positive correlation between Reviews & Installations.

The attributes were separated based on their nature into categorical and numeric attributes. We used the χ^2-test to identify the best categorical attributes. The feature scores for the categorical features are shown in Figure 3. The feature scores for the numerical attributes are shown in Figure 4.

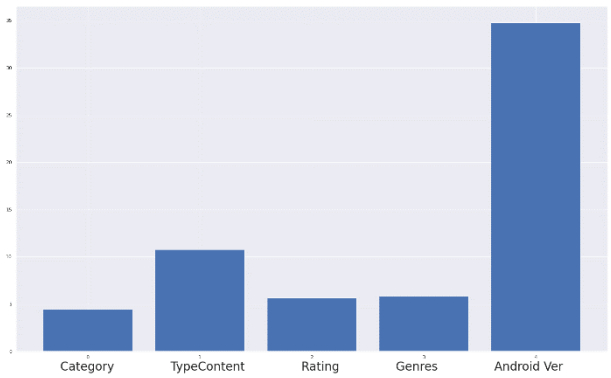

Fig. 3. Feature Importance for Categorical Attributes

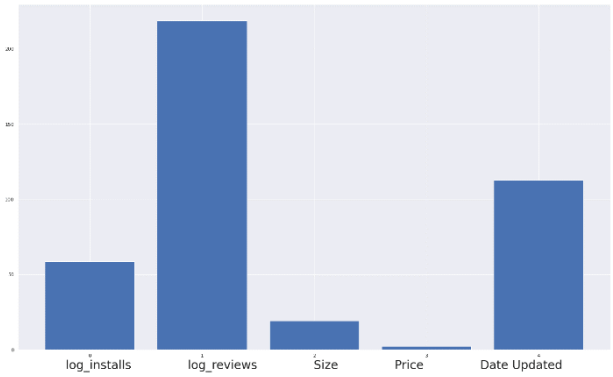

Fig. 4. Feature importance for numerical attributes

Analysis of Variance (ANOVA) was performed to identify the best numeric attributes. Figure 5 shows the mutual information scores for all the attributes, based on which the most informative features were found to be – log_reviews > log_installs > genres > dates > category > android version > type > size > price > content rating.

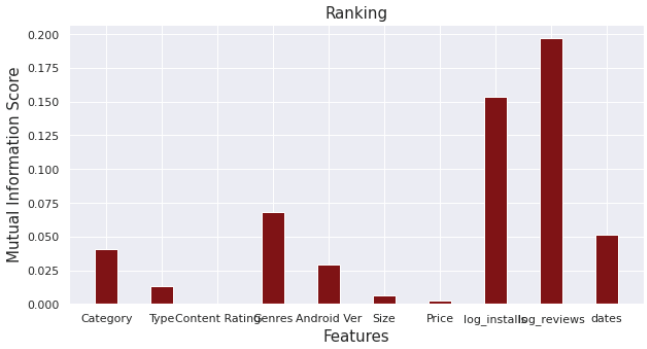

Fig. 5. Mutual Information Score

III. Machine Learning Regression Models

The train and test sets were split randomly in a 2:1 ratio, hence giving us 5490 data points for the train set and 2705 for the test set. We used Root Mean Squared Error (RMSE) as our accuracy metric, which is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{t}(Ra_i - Rp_i)^2}{t}}$$

t – total number of samples in the test set.

Ra – is the actual rating.

Rp – is the predicted rating.

We used various algorithms to train our model and predict the ratings on the test set. These algorithms along with their RMSE values are mentioned in Table 1.

TABLE I. ML Regression Algorithms and their RMSE values

Algorithm	RMSE
Gradient Boosting Regressor	7.4162
ElasticNet Regression	4.9038
Stochastic Gradient Descent Regressor	22.2709
Support Vector Regressor	5.1206
Bayesian Ridge Regression	25.3851
Kernel Ridge Regression	4.9196
Linear Regression	25.6584
XGBoost Regression	7.0834
Light Gradient Boosting Machine Regression	5.9221

From the table above we see that ElasticNet Regression gave us the best result followed by Kernel Ridge Regression. We tried to improve the accuracy by reducing the RMSE value using a few other ways as described in the subsequent sections.

IV. CatBoost Regression

Developed by Yandex Researchers and engineers, Categorical Boosting (CatBoost) is an algorithm for Gradient Boosting on Decision Trees [12], [13]. There are various advantages to using this technique. CatBoost handles categorical features automatically, using ordered target statistics. It also implements oblivious DTs, hence limiting the feature split per level to unity. This decreases the prediction time. Also, the default parameters are effective, which decreases the time required to tune the hyperparameters.

We performed hyperparameter tuning, using Grid Search, intending to minimize the same accuracy metric – RMSE. After obtaining the optimal values of hyperparameters, the model was trained & it gave us an RMSE of 0.67 on the test set.

V. Artificial Neural Network

We trained an artificial neural network to perform our regression task & see if the performance gets enhanced. The visualization of the proposed architecture of the ANN is shown in figure 6. It consists of four hidden layers of 12, 10, 8 & 4 neurons each. The input layer matches that of the input dimensionality of 6 neurons and the output layer with 1 neuron since it has to predict a real number. All the layers are densely connected with Rectified Linear Unit activation function in the first layer & the tan hyperbolic function in the hidden layers. All the All the layers had a normal kernel initializer.

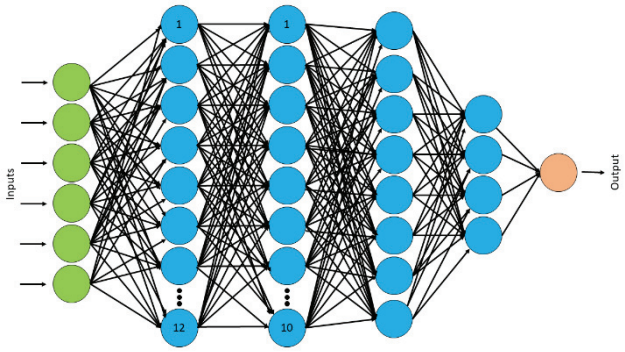

Fig. 6. Artificial Neural Network

Before feeding the data to the network, we performed min-max scaling to avoid any sort of data leakage. The data, in this case, was also split in a 2:1 ratio, hence giving us 5490 train samples & 2705 test samples. Figure 7 shows the model summary of the ANN architecture.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 12)                84

 dense_1 (Dense)             (None, 10)                130

 dense_2 (Dense)             (None, 8)                 88

 dense_3 (Dense)             (None, 4)                 36

 dense_4 (Dense)             (None, 1)                 5

=================================================================
Total params: 343
Trainable params: 343
Non-trainable params: 0
_________________________________________________________________
```

Fig. 7. The ANN Model Summary

The model was compiled intending to minimize the mean squared error loss using the Adam optimizer. The model was trained up to 15 epochs in batches of 32. The training loss with respect to increasing epochs has been shown in Figure 8.

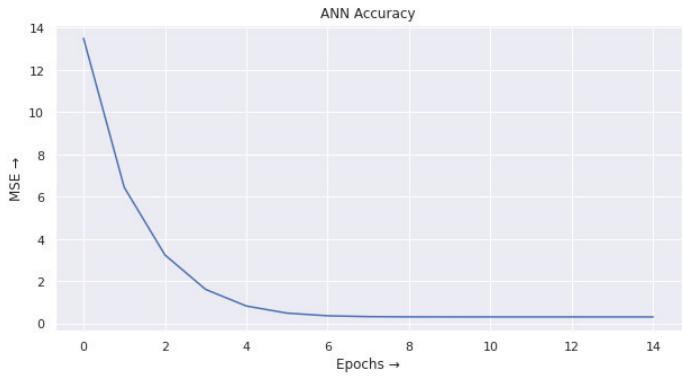

Fig. 8. Loss vs Epochs

This model gave us an MSE value of 24.24, hence RMSE = 0.492 on the test set.

VI. Results & Discussion

In addition to the ML regression techniques, the RMSE values for the CatBoost Algorithm and the MLP have been summarized in Table 2.

TABLE II. CatBoost & MLP along with their RMSE values

Algorithm	RMSE
CatBoost Regression	0.67
Multi-Layer Perceptron	0.49

The above-mentioned techniques outperformed the previously mentioned ML regression techniques. The Multi-layered Perceptron performed marginally better than the Categorical Boosting regression.

VII. Conclusion & Future Scope

As already mentioned by [3], the dataset is inadequate for model training & deployment, but keeping our target attributes intact helped us reduce the error significantly up to 0.49. Regression techniques are being developed by statisticians and mathematicians almost every year and one needs to be updated with them, what if any newly developed ML technique may also give us better results?

We expect that using certain LSTM architectures for regression and performing appropriate hyperparameter tuning in them might lead us to a further reduction in the RMSE values. This will be tested & demonstrated in our future works when we take another step ahead.

References

[1] A. Setiawan and V. C. Mawardi, "Android Application For Analysis Review On Google Playstore Using Support Vector Machine Method," 2022 5th International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2022, pp. 331-336, doi: 10.1109/ICOIACT55506.2022.9972122.

[2] B. Moharana, B. B. Biswal, S. Dey, M. K. Rath and S. Banerjee, "Play Store App Analysis & Rating Prediction using Classical ML Models & Artificial Neural Network," ICCUBEA 2023, in press.

[3] P. B. Prakash Reddy and R. Nallabolu, "Machine learning based Descriptive Statistical Analysis on Google Play Store Mobile Applications," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 647-655, doi: 10.1109/ICIRCA48905.2020.9183271.

[4] N. S. Chowdhury and R. R. Raje, "SERS: A Security-Related and Evidence-Based Ranking Scheme for Mobile Apps," 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), Los Angeles, CA, USA, 2019, pp. 130-139, doi: 10.1109/TPS-ISA48467.2019.00024.

[5] Y. Shi and D. Brown, "An Attempt to Discover Analytical Information for Multi-Dimensional Data Sets," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2018, pp. 1-5, doi: 10.1109/ICIRCA.2018.8597350.

[6] M. R. Islam, "Numeric rating of Apps on Google Play Store by sentiment analysis on user reviews," 2014 International Conference on Electrical Engineering and Information & Communication Technology, Dhaka, Bangladesh, 2014, pp. 1-4, doi: 10.1109/ICEEICT.2014.6919058.

[7] G. M. Muradul Bashir, M. S. Hossen, D. Karmoker and M. J. Kamal, "Android Apps Success Prediction Before Uploading on Google Play Store," 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2019, pp. 1-6, doi: 10.1109/STI47673.2019.9068071.

[8] R. M. Amir Latif, M. Talha Abdullah, S. U. Aslam Shah, M. Farhan, F. Ijaz and A. Karim, "Data Scraping from Google Play Store and Visualization of its Content for Analytics," 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2019, pp. 1-8, doi: 10.1109/ICOMET.2019.8673523.

[9] Qolbi, Shafira & Zahra, Aishaa & Larasati Anisa Rahma, Intan. (2022). ANALISIS DATASET GOOGLE PLAYSTORE MENGGUNAKAN METODE EXPLORATORY DATA ANALYSIS Analysis of Google Playstore Datasets Using Exploratory Data Analysis Methods.

[10] Prakash, G., & Koshy, J. (2021). Google Play Store Apps [Dataset]. In *Google Play Store App data of 2.3 Million+ applications.* (Version 7). Kaggle. https://www.kaggle.com/datasets/gauthamp10/google-playstore-apps.

[11] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

[12] CatBoost: unbiased boosting with categorical features, *Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, Andrey Gulin. NeurIPS, 2018*

[13] CatBoost: gradient boosting with categorical features support, *Anna Veronika Dorogush, Vasily Ershov, Andrey Gulin. Workshop on ML Systems at NIPS 2017*